



TUGAS AKHIR

**IMPLEMENTASI *ARTIFICIAL INTELLIGENCE* PADA
PERMAINAN *SNAKE* DENGAN
METODE *REINFORCEMENT LEARNING***

Sulistyawati Abdillah Rosyid
NIM. 11181079

Nisa Rizqiya Fadhliana, S.Kom., M.T.
Rizky Amelia, S.Si., M.Han.

Program Studi Informatika
Jurusan Matematika dan Teknologi Informasi
Institut Teknologi Kalimantan
Balikpapan, 2024

LEMBAR PERSETUJUAN

Proposal Tugas Akhir dengan judul:

**“IMPLEMENTASI *ARTIFICIAL INTELLIGENCE* PADA PERMAINAN
SNAKE DENGAN METODE *REINFORCEMENT LEARNING*”**

Yang disusun oleh:



Sulistyawan Abdillah Rosyid

NIM. 11181079

Telah diperiksa dan disetujui oleh dosen pembimbing:

Dosen Pembimbing Utama

Dosen Pembimbing Pendamping

Nisa Rizqiya Fadhliana, S.Kom., M.T.

NIP/NIPH. 198804102019032020

Rizky Amelia, S.Si., M.Han.

NIP/NIPH. 199308092022032016

KATA PENGANTAR

Puji syukur kepada Tuhan Yang Maha Esa atas berkat dan anugerah-Nya sehingga kami dapat menyelesaikan proposal tugas akhir yang berjudul:

“IMPLEMENTASI *ARTIFICIAL INTELLIGENCE* PADA PERMAINAN *SNAKE* DENGAN METODE *REINFORCEMENT LEARNING*”

Proposal tugas akhir ini merupakan salah satu syarat yang harus ditempuh untuk menyelesaikan Program Sarjana di Program Studi Informatika, Jurusan Matematika Teknologi dan Informasi, Institut Teknologi Kalimantan (ITK) Balikpapan. Untuk itu penulis mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Tuhan YME atas segala nikmat yang diberikan sehingga dapat terlaksana pembuatan laporan ini.
2. Ibu Nisa Rizqiya Fadhliana, S.Kom., M.T. selaku Dosen Pembimbing Utama dan Koordinator Program Studi Informatika Jurusan Matematika Teknologi dan Informasi ITK.
3. Ibu Rizky Amelia, S.Si., M.Han. selaku Dosen Pembimbing Pendamping dan Dosen Wali dari penulis.
4. Bapak Ibu Seluruh Dosen serta Tenaga Kependidikan Program Studi Informatika Jurusan Matematika dan Teknologi Informasi ITK.
5. Khoirun Nisa Al Fahmi selaku istri dari penulis dan keluarga dari penulis yang selalu suportif untuk menyelesaikan laporan ini.
6. Serta semua pihak yang terlibat dalam penyusunan proposal tugas akhir ini.

Penulis menyadari bahwa penyusunan laporan Tugas Akhir ini masih jauh dari sempurna, karena itu kami mengharapkan segala kritik dan saran yang membangun.

Semoga Tugas Akhir ini dapat bermanfaat bagi kita semua. Atas perhatiannya penulis ucapkan terima kasih.

Balikpapan, 10 September 2024

Penyusun

ABSTRAK

Penelitian ini membahas implementasi *Artificial Intelligence* (AI) pada permainan *Snake* menggunakan metode *Reinforcement Learning* (RL). Tujuan utama dari penelitian ini adalah mengembangkan agen yang mampu bermain *Snake* secara optimal melalui penerapan algoritma RL, khususnya *Q-Learning* dan *Deep Q-Network* (*DQN*). Lingkungan simulasi *Snake* dikembangkan untuk melatih agen AI dengan fokus pada interaksi agen dengan lingkungan, seperti menghindari dinding dan memakan makanan untuk memaksimalkan nilai *reward*. Analisis dilakukan untuk mengevaluasi kinerja agen berdasarkan parameter-parameter yang mempengaruhi, termasuk struktur *reward*, ukuran *grid*, dan kecepatan permainan. Hasil penelitian menunjukkan bahwa algoritma RL yang diterapkan mampu meningkatkan performa agen dalam memainkan *Snake*, dengan evaluasi performa yang dibandingkan dengan metode lain. Penelitian ini berkontribusi pada pengembangan RL yang lebih efisien dan memiliki potensi aplikasi lebih luas dalam robotika, navigasi otonom, dan sistem kompleks lainnya.

Kata Kunci: Artificial Intelligence, Reinforcement Learning, Q-Learning, Deep Q-Network, Snake

DAFTAR ISI

LEMBAR PERSETUJUAN.....	ii
KATA PENGANTAR	iii
ABSTRAK	v
DAFTAR ISI.....	vi
DAFTAR GAMBAR	viii
DAFTAR TABEL.....	ix
1. BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah.....	3
1.3 Tujuan Penelitian.....	3
1.4 Batasan Penelitian	3
1.5 Manfaat Penelitian.....	4
2. BAB 2 TINJAUAN PUSTAKA	6
2.1 <i>Artificial intelligence (AI)</i>	6
2.2 <i>Machine Learning</i>	7
2.3 <i>Reinforcement Learning</i>	8
2.4 Teori Permainan	10
2.5 Deep Learning	12
2.6 Penelitian Terdahulu.....	16
3. BAB 3 METODOLOGI PENELITIAN.....	23
3.1 Gambaran Besar Penelitian	23

3.2	Diagram Alir Penelitian.....	24
3.3	Prosedur Penelitian.....	25
3.3.1	Studi Literatur	25
3.3.2	Pengembangan Lingkungan Simulasi	25
3.3.3	Penerapan Algoritma Reinforcement Learning	26
3.3.4	Analisis Kinerja Agent <i>Artificial intelligence</i>	28
3.3.5	Pengujian dan Validasi Hasil	29
3.4	Jadwal Penelitian.....	30
DAFTAR PUSTAKA		32

DAFTAR GAMBAR

Gambar 2. 1 Skema Reinforcement Learning.....	9
Gambar 3. 1 Diagram Alir Penelitian	24
Gambar 3. 2 Skema Deep Q Network.....	27

ring

DAFTAR TABEL

Tabel 2.1 Penelitian Terdahulu	17
Tabel 3.2 Tabel Rencana Penelitian.....	30

Balajas menulis paragraf

BAB 1

PENDAHULUAN

panjang

Pada bab ini menjelaskan mengenai pendahuluan dari penelitian yang terdiri dari latar belakang, rumusan masalah, tujuan, manfaat dari dilakukannya penelitian ini.

X
Tidak boleh 1 halaman

1.1 Latar Belakang

Artificial intelligence (AI) merupakan sebuah istilah yang sering digaungkan akhir-akhir ini. Salah satu penyebabnya adalah saat ChatGPT dirilis oleh OpenAI pada November 2022. Sejak saat itu banyak produk-produk AI lain mulai bermunculan. Metode *Deep Learning* (DL), *Natural Language Processing* (NLP), *Computer Vision* banyak digunakan oleh perusahaan penghasil AI.

AI sendiri sebenarnya bukanlah sesuatu yang baru. Gagasan mengenai AI ini ditemukan pada tahun 1956, tetapi karena keterbatasan kala itu membuat gagasan AI ini menjadi sesuatu yang diremehkan oleh sebagian besar orang. Saat ini AI dibuat dengan tiga cara yaitu *Supervised Learning*, *Unsupervised Learning*, dan *Reinforcement Learning*.

Perkembangan teknologi komputer dan kecerdasan buatan (AI) telah mengalami kemajuan yang pesat dalam beberapa tahun terakhir. Salah satu cabang AI yang mendapatkan perhatian khusus adalah cabang *Machine Learning* (ML). Cabang ini mencakup beragam metode dan algoritma yang mengajarkan komputer untuk cara untuk mengambil keputusan dan tindakan. Dari beragam model *Machine Learning*, *Reinforcement Learning* (RL) cukup menonjol sebagai teknik yang menjanjikan untuk aplikasi di lingkungan yang dinamis dan kompleks. Dalam model ini sebuah agen belajar melalui interaksinya dengan lingkungan untuk memaksimalkan sebuah *reward*.

Permainan “Snake” adalah sebuah gim klasik yang ideal untuk mengembangkan algoritma *Reinforcement Learning* (RL) karena kesederhanaan dan

tantangan pada gim ini. Dalam gim ini, agen (Ular/*Snake*) diharuskan untuk mencari dan mendapatkan makanan yang muncul secara acak sembari menghindari agar tidak tertabrak dinding atau diri agen sendiri yang selalu bertumbuh setiap memakan sebuah makanan. Penerapan AI pada gim *Snake* menggunakan teknik RL memberikan beberapa kemungkinan, termasuk sebagai platform yang dapat diskalakan untuk menguji algoritma *Reinforcement Learning* dalam lingkungan yang terkendali dan dinamis. Aplikasi ini juga dapat disesuaikan dengan bidang lainnya seperti robotika dan navigasi. Gim ini memungkinkan proses pembelajaran dan adaptasi agen AI divisualisasikan dengan jelas. Hal ini memberikan wawasan penting tentang bagaimana algoritma *Reinforcement Learning* belajar dari pengalaman.

Pada jurnal sebelumnya milik Crespo dan Wichert. Dalam pembelajaran penguatan *reinforcement learning*, *Deep Q-Networks (DQN)* digunakan secara luas untuk mempelajari keputusan optimal dalam lingkungan permainan, seperti pada gim Atari. *DQN* memanfaatkan jaringan saraf untuk mendekati fungsi nilai yang memprediksi seberapa baik suatu tindakan dalam situasi tertentu. Peningkatan penting pada metode *DQN* meliputi *Averaged-DQN*, yang menstabilkan proses pembelajaran dengan mengurangi variabilitas dalam perkiraan *Q-values*, serta *Rainbow DQN*, yang menggabungkan beberapa perbaikan, termasuk *Double Deep Q-Learning*, dueling networks, dan prioritized replay untuk meningkatkan performa algoritma dalam berbagai tugas gim (Crespo & Wichert, 2020).

Penelitian ini bertujuan untuk mengembangkan agen yang mampu bermain *Snake* dengan sangat baik dan mengevaluasi efisiensi dan efektivitas berbagai algoritma RL yang diterapkan untuk permainan *Snake*. Selain itu, penelitian ini juga bertujuan untuk menemukan masalah dan peluang dalam penerapan RL dalam permainan *Snake* dan memberikan saran untuk penelitian tambahan. Penelitian ini diharapkan dapat meningkatkan pemahaman kita tentang *Reinforcement Learning* secara teoritis dan dalam aplikasinya yang kompleks dan dinamis. Ini juga akan membuka jalan untuk inovasi lebih lanjut dalam penggunaan AI untuk aplikasi yang kompleks dan dinamis.

→ Mana tanda tanya?

1.2 Perumusan Masalah

Berdasarkan latar belakang, maka didapatkan rumusan masalah sebagai berikut:

1. Bagaimana mengimplementasikan algoritma *Reinforcement Learning* untuk membuat sebuah agen AI yang mampu bermain gim “Snake” secara baik.
2. Apa saja parameter yang memengaruhi kinerja agen dalam permainan dan bagaimana pengaruhnya.
3. Bagaimana keefektifan algoritma *Reinforcement Learning* dibandingkan dengan metode lainnya

9
6

→ Cara ulhur?

1.3 Tujuan Penelitian

Sesuai dengan rumusan masalah, tujuan dari penelitian ini adalah sebagai berikut:

1. Mengimplementasikan algoritma *Reinforcement Learning* pada sebuah agen untuk gim “Snake” secara baik.
2. Mengetahui parameter yang memengaruhi kinerja agen dalam permainan dan bagaimana pengaruh dari parameter tersebut terhadap kinerja agen.
3. Membandingkan efektivitas dari algoritma *Reinforcement Learning* pada gim “Snake” dengan metode lainnya.

1.4 Batasan Penelitian

Adapun batasan masalah pada penelitian adalah sebagai berikut:

1. Algoritma yang digunakan adalah variasi dari jenis-jenis tertentu dari *Reinforcement Learning*, seperti *Q-Learning*, *Deep Q-Network* atau varian lain yang relevan.
2. Implementasi akan dilakukan dengan *framework* atau *library* tertentu seperti TensorFlow, PyTorch, atau OpenAI Gym.

9

buat model?

atau
menggunakan
model
yg
sudah
ada?

3. Pelatihan agen akan difokuskan pada permainan *Snake* standar dengan *grid* dua dimensi tanpa modifikasi aturan atau lingkungan permainan yang kompleks.
4. Analisis berfokus pada bagaimana variasi parameter ini mempengaruhi kinerja agen dalam hal skor rata-rata, panjang antrean, dan waktu bertahan hidup.
5. Parameter yang akan dianalisis meliputi ukuran *grid*, kecepatan permainan, struktur *reward system*, dan parameter internal algoritma *Reinforcement Learning* seperti *learning rate*, *discount factor*, dan *exploration rate*.
6. Kinerja agen yang diperoleh dari algoritma *Reinforcement Learning* dibandingkan dengan teknik dasar lainnya seperti pendekatan berbasis aturan dan algoritma pembelajaran terawasi sederhana.
7. Evaluasi didasarkan pada metrik kinerja seperti skor rata-rata yang dicapai oleh agen, panjang antrean maksimum yang dicapai, dan stabilitas kinerja.
8. Penelitian ini dibatasi pada evaluasi dalam lingkungan simulasi, tanpa pengujian dalam lingkungan fisik atau penerapan langsung di luar lingkup Permainan Ular.

→ Dimana?

1.5 Manfaat Penelitian

Adapun manfaat secara umum dari penelitian ini adalah memberikan wawasan lebih dalam tentang cara kerja algoritma pembelajaran penguatan dalam lingkungan dinamis berbasis aturan seperti Permainan *Snake*. Hasil penelitian ini dapat mengarah pada pengembangan dan peningkatan algoritma *Reinforcement Learning* yang lebih efisien dan efektif. Permainan Ular berfungsi sebagai studi kasus yang sangat baik untuk membuat prototipe aplikasi RL sebelum menerapkannya pada masalah dunia nyata yang lebih kompleks.

Peneliti yang berpartisipasi dalam penelitian ini akan mendapatkan pengalaman langsung menerapkan dan menguji algoritma AI serta mengembangkan keterampilan analisis data dan pemecahan masalah. Studi ini memungkinkan peneliti untuk mengembangkan keterampilan teknis dalam pemrograman, menggunakan *library* AI seperti TensorFlow dan PyTorch, dan menerapkan metode ilmiah. Selain itu, teknik

dan wawasan yang diperoleh dengan menerapkan *Reinforcement Learning* pada gim *Snake* dapat diterapkan pada berbagai bidang lainnya, seperti robotika, navigasi otonom, dan pengelolaan sumber daya.

Penelitian ini juga memiliki manfaat signifikan bagi industri permainan. Dengan memahami dan menerapkan algoritma RL yang lebih baik, pengembang permainan dapat menciptakan pengalaman bermain yang lebih menarik dan adaptif. Algoritma yang lebih efisien dapat digunakan untuk mengembangkan karakter non-pemain (NPC) yang lebih cerdas, meningkatkan interaksi pemain, dan menciptakan tantangan yang lebih dinamis dalam permainan. Selain itu, teknik yang diperoleh dari penelitian ini dapat membantu dalam pengembangan sistem rekomendasi yang lebih baik, yang dapat meningkatkan keterlibatan pemain dan retensi pengguna dalam permainan.

fishbone?

Analisis swot?

Game snake seperti apa
yg digunakan?

BAB 2

TINJAUAN PUSTAKA

Pada bab ini akan menjelaskan terkait tinjauan pustaka yang digunakan pada penelitian.

2.1 *Artificial intelligence (AI)*

Artificial intelligence (AI) adalah cabang ilmu komputer yang berfokus pada pengembangan sistem yang mampu melakukan tugas yang biasanya membutuhkan kecerdasan manusia, seperti pengenalan suara, pengambilan keputusan, dan terjemahan bahasa, AI didefinisikan sebagai studi tentang agen cerdas, yaitu entitas yang dapat merasakan lingkungan mereka dan mengambil tindakan untuk memaksimalkan peluang keberhasilan dalam mencapai tujuan tertentu. Dalam definisi ini, AI mencakup berbagai pendekatan, mulai dari logika simbolik hingga pembelajaran mesin yang lebih modern (Russell, 2020).

Dalam konteks pembelajaran mesin, melibatkan algoritma yang memungkinkan komputer belajar dari data dan memperbaiki performa mereka dari waktu ke waktu tanpa diprogram secara eksplisit. Pendekatan ini telah berkembang pesat dalam beberapa tahun terakhir dan diterapkan dalam berbagai bidang, termasuk kesehatan, transportasi, dan keuangan, untuk membuat prediksi yang lebih akurat dan otomatisasi proses yang kompleks (Goodfellow, dkk., 2016).

AI dapat dikelompokkan menjadi beberapa kategori utama berdasarkan kemampuannya, yaitu AI lemah (*narrow AI*) dan AI kuat (*general AI*). AI lemah dirancang untuk melakukan tugas tertentu dengan sangat baik, seperti pengenalan wajah atau pengolahan bahasa alami, tetapi tidak memiliki kemampuan untuk melakukan tugas di luar dari spesialisasinya. Sementara itu, AI kuat adalah sebuah konsep di mana mesin memiliki kecerdasan umum yang setara dengan manusia, dengan kemampuan untuk belajar dan beradaptasi dalam berbagai situasi. Meskipun

AI kuat masih merupakan tujuan jangka panjang yang belum tercapai, AI lemah sudah diterapkan luas dalam kehidupan sehari-hari (Tegmark, 2017).

Perkembangan AI yang pesat membawa potensi besar tetapi juga tantangan serius terkait etika, keamanan, dan dampak sosial. Bostrom mengemukakan bahwa ketika AI mencapai tingkat super intelligence, atau kecerdasan yang jauh melampaui kecerdasan manusia, ini dapat membawa perubahan yang sangat besar dalam peradaban manusia, termasuk peluang dan ancaman eksistensial. Oleh karena itu, Bostrom menekankan pentingnya penelitian untuk memastikan bahwa AI berkembang dengan cara yang aman dan bermanfaat bagi umat manusia (Bostrom, 2014).

2.2 *Machine Learning*

Belakangan ini Pembelajaran Mesin / *Machine Learning* sungguh banyak digunakan, bahkan bisa lebih banyak dari yang kita perkirakan. Contohnya saat kita mencari “Hadiah” untuk teman atau sahabat kita di internet. Kita akan mengetikkan sebuah kata kunci maka muncullah hal-hal yang berkaitan dengan kata kunci tersebut. Saat kita membuka kotak email kita lagi, tanpa disadari, kita akan menemukan sebuah email sejenis masuk ke kotak email kita atau mungkin dapat kita temukan pada *spam*. Selanjutnya saat pergi ke sebuah supermarket untuk membeli popok untuk dijadikan hadiah kepada sahabat. Saat ingin membayar, kasir memberikan sebuah kupon diskon untuk bir. Aksi kasir memberikan sebuah kupon diskon pada orang yang membeli popok ditunjukkan oleh *software* juga akan membeli bir. Data itulah yang dapat digunakan untuk mencari strategi untuk bisnis.

Apa yang akan kita cari pada sebuah data mentah itu tidak akan jelas. Contohnya dalam mendeteksi sebuah email *spam* mencari sebuah kata dalam email tentu tidak akan terlalu membantu. Bahkan akan terjadi kesalahan-kesalahan pendeteksian *email spam*. Karenanya akan lebih baik jika mencari gabungan dari beberapa kata yang ada dalam sebuah *email*, digabungkan dengan panjang *email*, laporan *spam* dari user lain, dan banyak hal lainnya. Dengan pertimbangan itu semua

akan menjadi jelas sebuah *email* adalah *spam* atau tidak. Inilah salah satu kegunaan *Machine Learning* untuk mengubah data mentah menjadi informasi yang dapat digunakan (Harrington, 2012).

Machine Learning adalah cabang dari kecerdasan buatan (*artificial intelligence*) yang berfokus pada pengembangan algoritma dan teknik yang memungkinkan komputer untuk belajar dari data. Dalam pengertian ini, mesin dapat belajar dari pengalaman atau data masa lalu untuk membuat prediksi atau keputusan tanpa harus diprogram secara eksplisit untuk setiap tugas (Mitchell, 1997).

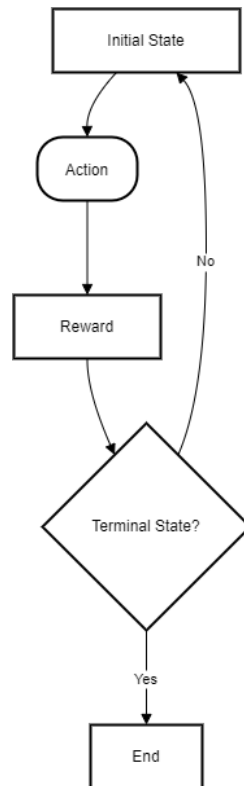
Machine Learning melibatkan proses di mana model atau algoritma "dilatih" dengan menggunakan data historis. Setelah proses pelatihan ini, model dapat digunakan untuk membuat prediksi atau menyelesaikan masalah tertentu berdasarkan data baru. Proses ini mencakup berbagai teknik, seperti *supervised learning*, *unsupervised learning*, dan *reinforcement learning*, yang masing-masing memiliki aplikasi dan pendekatan yang berbeda tergantung pada jenis data dan tujuan analisis (Alpaydin, 2020).

Machine Learning telah menjadi landasan utama dalam berbagai bidang, mulai dari pengenalan suara dan gambar hingga analisis prediktif dalam bisnis dan industri. *Machine Learning* memungkinkan sistem untuk mengenali pola kompleks dalam data dan secara otomatis memperbaiki kinerjanya seiring waktu, membuatnya menjadi alat yang sangat kuat dalam era data besar saat ini (Goodfellow, dkk., 2016).

2.3 *Reinforcement Learning*

Reinforcement Learning (RL) adalah suatu metode pembelajaran mesin di mana sebuah agen belajar untuk membuat keputusan dengan melakukan aksi-aksi di dalam lingkungan tertentu guna memaksimalkan suatu nilai *reward* kumulatif. *Reinforcement Learning* menekankan pada interaksi antara agen dan lingkungan melalui percobaan dan kesalahan, di mana agen tersebut mengamati status lingkungannya, memilih aksi, dan kemudian menerima feedback berupa *reward*.

Proses ini memungkinkan agen untuk belajar secara mandiri tanpa memerlukan data pelatihan yang lengkap seperti dalam *supervised learning* (Sutton & Barto, 2018).



Gambar 2. 1 Skema Reinforcement Learning

Dalam konteks ini, agen bertujuan untuk menemukan kebijakan (*policy*) yang menentukan bagaimana agen harus bertindak di berbagai situasi agar mendapatkan *reward* maksimal. *Reinforcement Learning* sering kali digunakan dalam skenario di mana keputusan harus dibuat secara berurutan dan dampaknya mungkin tidak segera terlihat, seperti dalam permainan video gim atau pengendalian robotika (Mnih, dkk., 2015).

Selain itu, *reinforcement learning* memiliki berbagai aplikasi di dunia nyata yang semakin berkembang. *Reinforcement Learning* telah digunakan untuk

mengalahkan juara dunia dalam permainan Go, melalui sistem bernama AlphaGo. Sistem ini memanfaatkan teknik *Monte Carlo Tree Search* (MCTS) bersama dengan *deep neural networks* untuk mengevaluasi posisi dan memilih langkah terbaik. Keberhasilan ini menyoroti potensi RL dalam menangani masalah kompleks yang membutuhkan pengambilan keputusan strategis di berbagai domain, mulai dari permainan hingga industri otomotif dan keuangan (Silver, dkk., 2016).

Penggunaan Reinforcement Learning juga mulai terlihat dalam pengoptimalan proses bisnis dan industri. Pada sektor energi, Reinforcement Learning digunakan untuk mengoptimalkan pengelolaan jaringan listrik pintar (*smart grid*) dan pengaturan sumber daya energi terbarukan. Agen Reinforcement Learning dapat dilatih untuk merespon perubahan permintaan dan penawaran listrik, serta memprediksi konsumsi energi untuk meningkatkan efisiensi dan stabilitas sistem (François-Lavet, dkk., 2018).

Secara umum, Reinforcement Learning terus berkembang seiring dengan peningkatan kemampuan komputasi dan kemajuan algoritma pembelajaran. Tantangan utama dalam Reinforcement Learning meliputi eksplorasi yang efisien, generalisasi terhadap situasi baru, dan pengurangan kompleksitas komputasi. Namun, dengan perkembangan teknik seperti *deep reinforcement learning*, *multi-agent Reinforcement Learning*, dan *Meta Reinforcement Learning*, pendekatan ini memiliki potensi yang besar untuk diterapkan dalam berbagai disiplin ilmu dan industri (Li, 2018).

2.4 Teori Permainan

Teori permainan adalah cabang matematika yang mempelajari interaksi strategis antara agen rasional, di mana hasil yang dicapai tergantung pada keputusan yang diambil oleh semua pemain. Teori ini umumnya diterapkan pada situasi kompetitif dan kooperatif yang melibatkan pilihan strategis dari berbagai pemain. Dalam konteks permainan *Snake*, kita dapat menggunakan beberapa prinsip dasar dari teori permainan untuk menganalisis pola perilaku dan strategi optimal yang memungkinkan pemain untuk mencapai hasil terbaik.

→ Apa hubungannya dgn game snake?

Permainan *Snake* adalah permainan pemain tunggal di mana pemain mengendalikan ular yang bergerak di dalam arena untuk mengambil / memakan buah yang muncul di layar. Setiap kali ular memakan buah, panjang tubuhnya bertambah, sehingga membuat permainan lebih sulit karena pergerakan menjadi lebih terbatas. Pemain akan kalah jika ular menabrak dirinya sendiri atau dinding.

Dari perspektif teori permainan, *Snake* dapat dianggap sebagai permainan sekuensial deterministik, di mana setiap langkah pemain berpengaruh terhadap posisi masa depan ular, mengingat panjang tubuh ular yang bertambah dari waktu ke waktu. Dalam teori permainan, ini mirip dengan extensive form games, di mana pemain membuat keputusan secara bertahap dengan informasi lengkap tentang kondisi permainan pada setiap langkah.

Pada permainan *Snake*, pendekatan umum yang sering digunakan pemain adalah strategi serakah (*greedy strategy*), di mana pemain berusaha segera menuju buah yang tersedia dalam langkah tersingkat. Meskipun strategi ini efisien untuk tahap awal, permainan menjadi lebih kompleks seiring dengan bertambahnya panjang ular. Dalam teori permainan, strategi serakah tidak selalu optimal karena bisa menyebabkan kebuntuan di masa depan. Permainan *Snake* juga dapat dilihat sebagai permainan yang mengandalkan heuristik, yaitu aturan-aturan yang digunakan pemain untuk membuat keputusan cepat. Contoh heuristik dalam *Snake* adalah selalu menjaga agar ular bergerak dalam pola melingkar atau persegi agar tetap memiliki ruang gerak yang cukup. Dalam hal ini, teori permainan berbicara tentang bagaimana heuristik membantu dalam membuat keputusan optimal di bawah tekanan waktu dan kondisi yang terbatas.

Dari perspektif ilmu komputer dan teori permainan, strategi optimal dalam *Snake* sering kali dikembangkan melalui pendekatan algoritmis. Beberapa algoritma yang relevan dengan analisis teori permainan seperti algoritma *Dijkstra* (*A-star*) dan *Hamiltonian Cycle*.

Dalam permainan ini, *reward* ular (agen) dalam memperoleh buah berbanding terbalik dengan risiko bertambahnya panjang tubuh yang dapat menyebabkan kekalahan. Setiap kali pemain memperoleh keuntungan (buah), tantangan juga

meningkat, sehingga menghasilkan keseimbangan antara risiko dan *reward*, sebuah konsep yang umum dalam teori permainan. Teori permainan sering kali melibatkan manajemen risiko, di mana pemain harus memilih antara strategi berisiko tinggi dengan *reward* besar atau strategi konservatif dengan hasil yang lebih terprediksi. Dalam *Snake*, pemain yang terus-menerus menggunakan strategi agresif untuk langsung mendapatkan buah mungkin lebih cepat gagal, sementara pemain yang lebih berhati-hati dapat bertahan lebih lama dengan strategi yang lebih defensif, seperti menjaga ruang gerak yang luas sebelum mengincar buah (Myerson, 1991).

2.5 Deep Learning

Deep Learning (DL) adalah teknik yang menggunakan jaringan saraf tiruan berlapis-lapis (deep neural networks) untuk menangani masalah non-linear yang kompleks. Pada kasus gim seperti *Snake*, RL digunakan agar agen (dalam hal ini, pemain *Snake*) belajar bagaimana bergerak dalam grid untuk memaksimalkan *reward*—biasanya dengan makan makanan dan menghindari benturan dengan tubuh atau dinding. Teori-teori utama yang ada pada Deep Reinforcement Learning (DRL) diantaranya adalah *Q-Learning*, Deep-Q-Network (*DQN*), Experience Replay, Target Network, Policy Gradient Method.

2.5.1. *Q-Learning*

Q-Learning bertujuan untuk menemukan kebijakan optimal yang memaksimalkan total *reward* jangka panjang. Pada dasarnya, *Q-Learning* menggunakan fungsi nilai aksi yang dikenal sebagai fungsi *Q* untuk mengestimasi seberapa baik tindakan tertentu yang dilakukan dalam suatu keadaan tertentu (Watkins, 1992).

Fungsi *Q* didefinisikan sebagai:

$$Q(s, a) = Q(s, a) + \alpha(r + \gamma \max_{a'} Q(s', a') - Q(s, a))$$

Keterangan:

$Q(s, a)$: Nilai *Q*
 s : State / Keadaan

Bagian RL dan deep

learning

Notasi
persamaan!

a	: Aksi
α	: <i>Learning Rate</i> / laju pembelajaran
r	: <i>Reward</i>
γ	: <i>Discount factor</i>
s'	: <i>State</i> Selanjutnya
a'	: Aksi selanjutnya
$\max_{a'} Q(s', a')$: Estimasi nilai maksimum Q untuk semua aksi (a') pada keadaan (s')

Nilai fungsi Q yang mewakili estimasi nilai dari suatu aksi a yang diambil pada keadaan s . *Learning rate* (laju pembelajaran), yaitu seberapa besar kita memperbarui nilai Q berdasarkan informasi baru. Nilai α biasanya berkisar antara 0 dan 1, di mana nilai yang lebih besar berarti pembelajaran lebih cepat, tetapi mungkin menyebabkan ketidakstabilan jika terlalu besar. *Reward* (hadiah) yang diterima setelah melakukan aksi a di keadaan s , dan pindah ke keadaan berikutnya s' . Ini adalah hadiah langsung yang diperoleh dari lingkungan. *Discount factor* (faktor diskon) menentukan seberapa besar pengaruh *reward* masa depan terhadap nilai Q saat ini. Nilai γ juga berkisar antara 0 dan 1, di mana 1 berarti *reward* masa depan dianggap sepenuhnya, dan 0 berarti hanya *reward* saat ini yang diperhitungkan. Next state (keadaan berikutnya) yang dicapai setelah mengambil aksi a dari keadaan s . Estimasi nilai maksimum Q untuk semua aksi a' yang bisa diambil pada keadaan berikutnya s' . Maksudnya, dari semua aksi yang mungkin, kita memilih aksi terbaik yang memaksimalkan nilai Q di keadaan s' (Sutton & Barto, 2018).

2. 5. 2. Deep-Q-Network (DQN)

Banyak aplikasi dari *reinforcement learning* seperti mobil yang dapat menyetir sendiri, Atari, konsumen marketing, pelayanan kesehatan, edukasi dan lainnya memiliki banyak *state* (s) dan atau aksi (a). Bahkan ada bisa saja ada beberapa *state* yang tidak dijumpai saat model training dijalankan. Hal ini

membutuhkan model representatif seperti *reward*, nilai dari *state-action* (Q), nilai dari *state* (V), dan peraturan yang dapat memunculkan beberapa *state* dan aksi. Agen mempelajari VFA (*Value Function Approximation*) dengan memperbarui bobot setiap langkah waktu berdasarkan *Temporal Difference* (TD) return saat ini. Setelah pembaruan, TD return tersebut dibuang, dan TD return baru digunakan untuk pembaruan bobot berikutnya. Metode ini tidak memaksimalkan penggunaan data yang telah diambil untuk menyesuaikan fungsi.

Istilah "Deep *Q-Learning*" digunakan karena jaringan saraf biasanya diimplementasikan dengan deep neural networks (DNNs) dalam banyak aplikasi. TD (*Temporal Difference*) adalah metode dalam *reinforcement learning* yang digunakan untuk memperkirakan nilai (*value*) dari suatu keadaan atau aksi berdasarkan perbedaan temporal, yaitu perbedaan antara nilai yang diprediksi saat ini dan nilai yang sebenarnya diterima di langkah berikutnya. TD menggabungkan aspek dari *Monte Carlo* dan *dynamic programming*.

Inti dari metode TD adalah TD error, yang mengukur kesalahan prediksi antara nilai saat ini dan estimasi nilai berikutnya. Pembaruan dilakukan secara bertahap setiap kali agen mengambil tindakan dan menerima umpan balik (*reward*). *TD learning* sering digunakan dalam algoritma seperti *Q-Learning* dan *SARSA*.

Berikut adalah contoh sederhana dari Update Temporal Difference:

$$V(s) \leftarrow V(s) + \alpha[r + \gamma V(s') - V(s)]$$

Keterangan:

$V(s)$: Nilai keadaan (s)
 α : Laju pembelajaran
 r : *Reward* yang diterima saat perpindahan s ke s'
 γ : Faktor diskon
 $V(s')$: Nilai prediksi keadaan selanjutnya s'

nomor persamaan nya mana?

Dibuat paragraf

Dengan menggunakan pendekatan ini, agen belajar dari pengalaman tanpa perlu menunggu episode selesai, berbeda dengan metode *Monte Carlo* yang memerlukan penyelesaian episode penuh untuk memperbarui nilai (Kuang, 2021).

2. 5. 3. Experience Replay

Untuk mengurangi korelasi, kita menyimpan dataset dari pengalaman sebelumnya ke dalam *replay buffer*. Data yang berbentuk tuple secara bertahap ditambahkan ke buffer saat agen berinteraksi dengan lingkungan. Implementasi paling sederhana adalah buffer dengan ukuran tetap, di mana data baru ditambahkan ke akhir buffer, sehingga mendorong data pengalaman yang paling lama keluar.

Pengambilan sampel dari sebagian kecil tuple untuk melatih jaringan saraf disebut *experience replay*. Selain mengurangi korelasi, *experience replay* memungkinkan kita belajar lebih banyak dari satu set tuple berkali-kali, dan secara umum memaksimalkan penggunaan pengalaman.

Untuk melakukan *experience replay*, kita secara acak mengambil sampel *batch* dari tuple pengalaman, menghitung nilai target, dan kemudian menggunakan *Stochastic Gradient Descent* (SGD) untuk memperbarui bobot jaringan saraf. Metode pelatihan jaringan saraf umum dapat digunakan untuk tujuan ini (Kuang, 2021).

2. 5. 4. Policy Gradient Method

Metode *policy gradient* mempelajari kebijakan secara langsung, sementara metode seperti TD, SARSA, dan *Q-Learning* adalah metode berbasis nilai, yang berfokus pada evaluasi dan pembaruan nilai dari tindakan atau keadaan tertentu. Teorema *policy gradient* adalah konsep penting dalam metode *policy gradient*, yang akan dijelaskan dengan bukti terperinci. Sebelum itu, dijelaskan contoh sederhana berupa MDP satu langkah, di mana proses dimulai dari

keadaan tertentu dan selesai setelah satu langkah dengan pemberian *reward* $r = R_{s,a}$.

Fungsi objektifnya adalah:

$$J(\theta) = E_{\pi_\theta}[r] = \sum_{s \in S} d(s) \sum_{a \in A} \pi_\theta(s, a) R_{s,a}$$

Nomor

Keterangan:

$J(\theta)$: Fungsi objektif *reward* yang akan dimaksimalkan oleh agent

$E_{\pi_\theta}[r]$: Nilai ekspektasi *reward*

$\sum_{s \in S}$: Penjumlahan keadaan s dalam himpunan S

$d(s)$: Distribusi keadaan

$\sum_{a \in A}$: Penjumlahan aksi a dalam himpunan A

$\pi_\theta(s, a)$: Kebijakan agent tergantung pada parameter θ

$R_{s,a}$: *Reward* yang diterima saat agen berada pada state s dan aksi a

Rumus ini menggambarkan cara menghitung ekspektasi *reward* total yang diharapkan (dinyatakan sebagai $J(\theta)$) dengan menjalankan kebijakan tertentu π_θ . Ekspektasi ini dihitung dengan menjumlahkan *reward* dari setiap pasangan keadaan-tindakan (s, a) , dikalikan probabilitas kebijakan untuk memilih tindakan tertentu pada suatu keadaan ($\pi_\theta(s, a)$), dan frekuensi terjadinya $d(s)$. Policy gradient theorem pada dasarnya mencoba untuk mengoptimalkan kebijakan dengan mengubah parameter θ untuk memaksimalkan *reward* $J(\theta)$ yang akan dihasilkan (Kuang, 2021).

2.6 Penelitian Terdahulu

Berdasarkan literatur yang telah digunakan pada penulisan proposal ini, bisa disimpulkan bahwa dengan adanya sebuah sistem informasi geografis dapat membantu

Cara ulus gimana ?

Mention tabel

pengguna untuk mengambil keputusan. Berikut ini adalah acuan yang memiliki keterkaitan dengan penelitian yang akan dilakukan.

Tabel 2.1 Penelitian Terdahulu

No	Nama Penulis dan Tahun Publikasi	Hasil
1	(Bakar, dkk., 2023)	Judul: <i>Fusion Sparse And Shaping Reward Function In Soft Actor-Critic Deep Reinforcement Learning For Mobile Robot Navigation</i> Metode: <i>Soft Actor Critic</i> Hasil: Saat ini, kemajuan dalam robot otonom sangat dipengaruhi oleh perkembangan teknologi terbaru. <i>Deep Reinforcement Learning</i> (DRL) memungkinkan sistem beroperasi secara otomatis, di mana robot mempelajari gerakan berdasarkan interaksi dengan lingkungan. Salah satu pendekatan DRL terbaru adalah <i>Soft Actor Critic</i> (SAC) yang mampu mengontrol tindakan berkelanjutan untuk menghasilkan gerakan yang lebih akurat. Meskipun SAC tangguh terhadap ketidakpastian, terdapat kelemahan dalam proses eksplorasi yang mempengaruhi kecepatan pembelajaran. Penelitian ini mengusulkan penggunaan fungsi ganjaran berbasis ganjaran jarang dan terstruktur dalam metode SAC untuk meningkatkan efektivitas pembelajaran robot. Hasil eksperimen menunjukkan bahwa kombinasi ini berhasil menavigasi robot ke sasaran dengan

Header Tabel

		meningkatkan akurasi hingga rata-rata kesalahan 4,99%.
2	(Ridho dkk., 2024)	<p>Judul: Penerapan <i>Deep Reinforcement Learning</i> dalam <i>Business Intelligence</i></p> <p>Metode: <i>Deep Neural Network</i></p> <p>Hasil: <i>Business Intelligence</i> (BI) adalah kombinasi alat seperti gudang data, OLAP, dan dasbor untuk analisis data mendalam. Gudang data mengumpulkan data akurat dari berbagai sumber, sementara OLAP memungkinkan analisis multidimensi secara real-time. <i>Deep Reinforcement Learning</i> (DRL) diidentifikasi sebagai teknik yang menjanjikan untuk mengatasi tantangan pengambilan keputusan berurutan dengan ketidakpastian. <i>Deep neural network</i> (DNN) juga digunakan dalam BI untuk prediksi yang lebih baik. Temuan menunjukkan bahwa <i>Double Deep Q Learning</i> dapat meningkatkan BI dengan mengoptimalkan sumber daya dan mengurangi waktu kalkulasi dalam masalah skala besar serta memodelkan pola strategi lawan dalam sistem multi-agen.</p>
3	(Afriyadi & Utomo, 2023)	<p>Judul: <i>Towards Human-Level Safe Reinforcement Learning in Atari Library</i></p> <p>Metode: <i>Double Deep Q Learning</i></p> <p>Hasil: Penelitian ini mengusulkan pendekatan baru untuk mengintegrasikan pembatasan keselamatan yang dibuat secara manual dalam algoritme <i>Double Deep Q-Network</i> (DDQN).</p>

Hender Tabal

Tujuannya adalah agar agen RL dapat belajar untuk bertindak dengan aman tanpa mengorbankan kinerja. Penelitian ini menggunakan pustaka Atari, khususnya gim *Super Mario Bros*, sebagai contoh lingkungan yang kompleks untuk melatih agen RL. Hasil penelitian menunjukkan bahwa metode yang diusulkan, termasuk pembatasan keselamatan, mampu mengungguli algoritme RL konvensional dalam hal keamanan dan pencapaian hasil yang optimal.

-
- | | | |
|---|--------------------|---|
| 4 | (Taqwa dkk., 2023) | <p>Judul: <i>Designing A WSNs-based Smart Home Monitoring System through Deep Reinforcement Learning</i></p> <p>Metode: <i>Deep Reinforcement Learning</i></p> <p>Hasil: pengembangan sistem pemantauan rumah pintar berbasis <i>wireless sensor networks</i> (WSNs) dan <i>deep reinforcement learning</i> (DRL) untuk memantau parameter lingkungan seperti suhu, kelembapan, dan kadar CO₂. Sistem ini dirancang untuk memberikan rekomendasi apakah lingkungan rumah dalam kondisi "nyaman" atau "tidak nyaman." Proses eksperimen dilakukan melalui validasi akurasi sensor WSNs, pemilihan model DRL terbaik, dan pengujian sistem pemantauan. Hasil pengujian menunjukkan bahwa sistem ini memiliki tingkat akurasi tinggi (95.52% untuk DRL, 98.52% untuk WSNs) dan biaya rendah, serta mudah diimplementasikan. Sistem</p> |
|---|--------------------|---|
-

Header Tabel

ini juga dilengkapi aplikasi berbasis Android untuk memudahkan pengguna dalam memantau kondisi rumah secara *real-time*.

5 (Hidayat dkk., 2023)

Judul: *Modified Q-Learning Algorithm for Mobile Robot Real-Time Path Planning using Reduced States*

Metode: *Q-Learning*

Hasil: Perencanaan jalur adalah algoritme penting dalam robot otonom, termasuk robot pertanian. Salah satu metode *reinforcement learning* yang digunakan untuk perencanaan jalur robot adalah algoritme *Q-Learning*. Namun, metode *Q-Learning* konvensional membutuhkan biaya komputasi tinggi karena mengeksplorasi semua kemungkinan keadaan robot untuk menemukan jalur optimal. Studi ini memodifikasi algoritme *Q-Learning* dengan menghilangkan area yang tidak dapat dilalui sehingga area tersebut tidak dihitung sebagai grid yang harus diproses. Algoritme *Q-Learning* yang dimodifikasi ini diuji pada robot otonom di Agribusiness and Technology Park (ATP), IPB. Hasil simulasi menunjukkan bahwa modifikasi ini mengurangi biaya komputasi sebesar 50,71%, dengan waktu rata-rata 25,74 detik dibandingkan dengan 50,75 detik pada metode *Q-Learning* asli. Meskipun begitu, kedua metode menghasilkan jumlah keadaan yang sama untuk jalur optimal, yaitu 56 keadaan. Algoritme *Q-Learning* yang dimodifikasi juga dapat

Header Tabel

		menemukan jalur dengan nilai parameter learning rate minimal 0,2 ketika faktor diskon bernilai 0,9.
6	(Zhu & Zhang, 2021)	<p>Judul: <i>Deep Reinforcement Learning Based Mobile Robot Navigation: A Review</i></p> <p>Metode: <i>Deep Q Network</i></p> <p>Hasil: Aplikasi pelaporan dan pelacakan kejahatan berhasil dibuat. Masyarakat dapat melaporkan pelaku kejahatan secara <i>realtime</i> melalui <i>smartphone</i> android. Kejahatan akan cepat ditangani menggunakan konsep <i>geofencing</i>. Di dalam aplikasi ini memiliki 2 jenis <i>user</i> yaitu masyarakat dan polisi, dengan adanya fitur <i>near me</i> masyarakat dan polisi dapat melihat apa yang terjadi di sekitarnya. Masyarakat harus <i>login</i> terlebih dahulu untuk menggunakan fitur <i>panic button</i>.</p>
7	(Alrahma dkk., 2024)	<p>Judul: <i>Application of Q-Learning Method for Disaster Evacuation Route Design Case Study: Digital Center Building UNNES</i></p> <p>Metode: <i>Q-Learning</i></p> <p>Hasil: Penelitian ini menerapkan <i>Q-Learning</i>, sebuah teknik pembelajaran penguatan, untuk merancang rute evakuasi secara efisien di lantai pertama Gedung Digital Center di UNNES, yang menghasilkan jalur optimal dengan akurasi tinggi dan efisiensi komputasi yang baik, bahkan ketika ada hambatan tambahan.</p>

Header Tabel

8	(Changbao dkk., 2024)	<p>Judul: <i>Research on Game Character Behavior Decision-Making System Based on Deep Reinforcement Learning</i></p> <p>Metode: <i>Deep Reinforcement Learning</i></p> <p>Hasil: Makalah ini mengeksplorasi penggunaan <i>Deep Reinforcement Learning</i> (DRL) dalam meningkatkan kemampuan pengambilan keputusan karakter gim. Metode tradisional seperti Pohon Perilaku dan Mesin Keadaan Terbatas memiliki keterbatasan dalam adaptabilitas dan kinerja karena sifatnya yang berbasis aturan. Studi ini menyajikan model DRL, khususnya menggunakan <i>Deep Q-Network</i> (DQN), yang menunjukkan peningkatan kualitas pengambilan keputusan dan kecepatan respons untuk karakter gim. Penelitian ini membahas penerapan DRL dalam lingkungan gim yang kompleks, menekankan keunggulannya seperti strategi perilaku adaptif dan pengalaman bermain gim yang lebih baik. Meskipun memiliki potensi, DRL menghadapi tantangan seperti waktu pelatihan yang panjang dan interpretabilitas model yang buruk</p>
---	-----------------------	--

BAB 3

METODOLOGI PENELITIAN

→ I belum aja?

Pada bab ini, dijelaskan mengenai metodologi yang akan digunakan pada penelitian ini, pengerjaan penelitian ini berisi garis besar penelitian, diagram alir penelitian, prosedur penelitian, dan rencana jadwal penelitian dalam **IMPLEMENTASI ARTIFICIAL INTELLIGENCE PADA PERMAINAN SNAKE DENGAN METODE REINFORCEMENT LEARNING**!!

3.1 Gambaran Besar Penelitian

Gambaran besar penelitian ini adalah memberikan panduan umum mengenai alur penelitian dari awal hingga akhir. Pengerjaan "Implementasi *Artificial intelligence* pada Permainan *Snake* dengan Metode Reinforcement Learning" dibagi menjadi 4 tahap.

Tahap pertama adalah pengembangan lingkungan simulasi permainan *Snake* yang akan digunakan sebagai platform pelatihan untuk agen AI. Pada tahap ini, dilakukan pemodelan permainan *Snake* dengan mempertimbangkan semua variabel lingkungan, seperti ukuran papan permainan, posisi makanan, dan kondisi-kondisi permainan yang memungkinkan. Simulasi ini dirancang agar agen dapat berinteraksi dengan lingkungan secara efektif untuk mempelajari tindakan yang menghasilkan *reward* optimal.

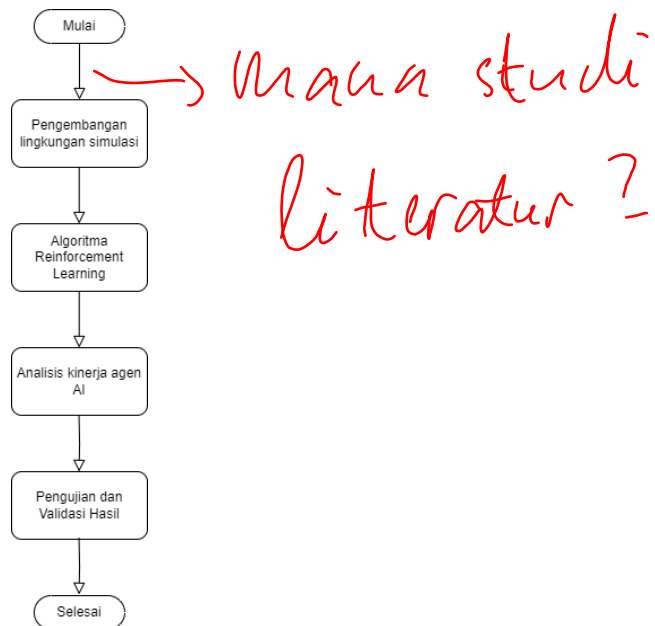
Tahap kedua adalah penerapan algoritma Reinforcement Learning, seperti *Q-Learning* dan *Deep Q-Network (DQN)*, pada lingkungan simulasi tersebut. Agen AI dilatih dengan berbagai iterasi permainan, di mana setiap tindakan yang diambil menghasilkan *reward* atau penalti. Pada tahap ini, algoritma *Q-Learning* akan menggunakan tabel nilai (Q-table) untuk menyimpan *reward* potensial dari setiap tindakan, sementara metode *DQN* akan memanfaatkan jaringan saraf tiruan untuk mengestimasi nilai Q dalam ruang tindakan yang lebih kompleks.

Tahap ketiga adalah analisis kinerja agen AI yang dilatih menggunakan algoritma-algoritma tersebut. Setelah agen selesai dilatih, dilakukan evaluasi terhadap kemampuan agen dalam memainkan permainan *Snake*, meliputi seberapa baik agen dapat menghindari tabrakan, memakan makanan, dan memaksimalkan skor. Hasil evaluasi ini akan menunjukkan efektivitas algoritma Reinforcement Learning yang digunakan.

Tahap keempat adalah pengujian dan validasi hasil. Pada tahap ini, dilakukan analisis konsistensi kinerja agen dan perbandingan antara hasil pelatihan menggunakan *Q-Learning* dan *DQN*. Diperiksa apakah hasil yang diperoleh konsisten dalam berbagai kondisi lingkungan permainan yang berbeda dan apakah agen menunjukkan kemampuan adaptasi yang baik terhadap perubahan. Hasil dari tahap ini akan menjadi dasar untuk menarik kesimpulan mengenai efektivitas metode Reinforcement Learning dalam permainan *Snake* dan potensi pengembangannya untuk aplikasi lain.

3.2 Diagram Alir Penelitian

Adapun diagram alir dari penelitian ini adalah sebagai berikut:



Gambar 3. 1 Diagram Alir Penelitian

Kurang-bata "Adapun" dan

Pada Gambar 3.1 ditunjukkan diagram alir dari penelitian ini. Terdapat 4 tahapan, yaitu Pengembangan lingkungan simulasi, Algoritma Reinforcement Learning, Analisis kinerja agen AI, dan pengujian dan validasi hasil.

" Dalam "

3.3 Prosedur Penelitian

Adapun penjelasan lebih lengkap terkait tahapan-tahapan yang ada pada sub bab diagram alir penelitian akan dijelaskan pada sub bab prosedur penelitian ini. Berikut penjelasan tahap-tahap yang dilakukan untuk menyelesaikan penelitian implementasi *artificial intelligence* pada permainan *snake* dengan metode reinforcement learning.

3.3.1 Studi Literatur

Studi literatur ini terdapat pada tahap pertama untuk mengumpulkan literasi yang sesuai dengan topik penelitian dalam implementasi *artificial intelligence* pada permainan *snake* dengan metode *reinforcement learning*. Beberapa literatur yang akan digunakan adalah tentang metode Double Q Network, penjelasan mengenai agen AI.

Adapun hasil dari tahap studi literatur adalah didapatkan penjelasan terkait metode, dan didapatkan informasi terkait dengan metode yang tepat serta kelebihan dari metode tersebut dalam proses pengembangan sistem model AI dengan *reinforcement learning*. Kemudian memberikan gambaran terkait tahapan metodologi yang harus dilalui berdasarkan metode yang digunakan.

3.3.2 Pengembangan Lingkungan Simulasi

Dalam tahap pengembangan lingkungan simulasi, kami akan membuat model permainan *Snake* yang komprehensif dengan mempertimbangkan berbagai variabel penting seperti ukuran papan permainan, posisi makanan, dan kondisi permainan yang beragam. Simulasi ini dirancang untuk memungkinkan agen kecerdasan buatan (AI) berinteraksi secara efektif dengan lingkungan, sehingga dapat mempelajari tindakan

yang menghasilkan *reward* optimal. Dengan lingkungan simulasi yang realistis dan dinamis, kami bertujuan untuk menciptakan platform pelatihan yang ideal bagi agen AI untuk mengembangkan strategi bermain yang efisien dan adaptif.

Agar simulasi ini dapat mendukung proses pembelajaran agen AI dengan maksimal, kami akan memastikan setiap elemen permainan *Snake* disimulasikan secara akurat. Ini termasuk penyesuaian tingkat kesulitan permainan dan penyediaan umpan balik yang tepat untuk setiap tindakan yang dilakukan oleh agen. Selain itu, simulasi ini akan dioptimalkan untuk memungkinkan iterasi yang cepat, memungkinkan agen untuk belajar dari pengalaman dengan cepat dan efektif.

Kami juga akan memasukkan fitur *logging* dan monitoring untuk memantau interaksi agen dengan lingkungan secara *real-time*. Fitur ini akan memberikan data yang diperlukan untuk analisis lebih lanjut dan penyesuaian strategi pembelajaran, sehingga agen dapat meningkatkan kinerjanya secara progresif. Dengan pendekatan ini, diharapkan agen AI dapat mempelajari kebiasaan bermain yang lebih cerdas dan efektif dalam waktu singkat.

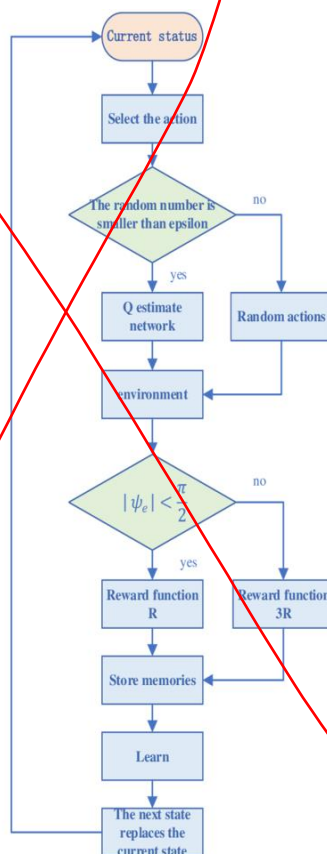
3.3.3 Penerapan Algoritma Reinforcement Learning

Pada tahap penerapan algoritma dalam penelitian "Implementasi *Artificial intelligence* pada Permainan *Snake* dengan Metode *Reinforcement Learning*", beberapa langkah penting dilakukan untuk memastikan agen AI dapat belajar dan beradaptasi dengan lingkungan permainan secara efektif.

Algoritma Reinforcement Learning yang dipilih untuk diterapkan adalah *Q-Learning* dan *Deep Q-Network (DQN)*. *Q-Learning* menggunakan tabel nilai (*Q-table*) untuk menyimpan *reward* potensial dari setiap tindakan, sedangkan *DQN* memanfaatkan jaringan saraf tiruan untuk mengestimasi nilai *Q* dalam ruang tindakan yang lebih kompleks. Algoritma tersebut diterapkan pada lingkungan simulasi yang telah dikembangkan. Proses ini melibatkan pengkodean algoritma dan integrasinya dengan simulasi permainan *Snake*, sehingga agen dapat mulai belajar dari interaksi

dengan lingkungan. Agen AI dilatih melalui berbagai iterasi permainan. Setiap tindakan yang diambil oleh agen menghasilkan *reward* atau penalti, yang digunakan untuk memperbarui strategi agen. Proses ini memungkinkan agen untuk belajar dari pengalaman dan meningkatkan kemampuannya dalam bermain *Snake*. Selama pelatihan, dilakukan pengujian untuk memantau kinerja agen dan menyesuaikan parameter algoritma jika diperlukan. Hal ini penting untuk memastikan bahwa agen dapat mencapai performa optimal dalam permainan.

Skema yang akan digunakan adalah sebagai berikut:



Paragraf?

Gambar 3. 2 Skema Deep Q Network

Diagram alur di atas menjelaskan proses pembelajaran reinforcement learning dalam permainan *Snake*. Proses dimulai dengan status atau keadaan saat ini dari permainan. Sistem kemudian memilih aksi yang akan diambil berdasarkan kebijakan

yang ada, yang dipengaruhi oleh nilai epsilon. Jika angka acak yang dihasilkan lebih kecil dari epsilon, aksi tersebut dipilih berdasarkan prediksi jaringan estimasi Q. Jika tidak, aksi dipilih secara acak.

Setelah aksi dipilih, lingkungan permainan diperbarui sesuai dengan aksi tersebut. Selanjutnya, sistem mengevaluasi apakah $|\psi_e| < \frac{\pi}{2}$. Jika ya, fungsi *reward* R diterapkan, memberikan reward sesuai dengan tindakan yang diambil. Jika tidak, fungsi *reward* 3R digunakan, yang mungkin memberikan penalti atau *reward* berbeda.

Memori dari setiap langkah disimpan untuk pembelajaran di masa depan. Sistem memanfaatkan memori ini untuk memperbarui jaringan Q, meningkatkan kemampuan pengambilan keputusan. Status baru menggantikan status saat ini, dan proses diulang untuk langkah berikutnya. Dengan demikian, agen terus belajar dan menyesuaikan tindakan untuk memaksimalkan fungsi hadiah, menyeimbangkan eksplorasi dan eksploitasi.

3.3.4 Analisis Kinerja Agent Artificial intelligence

Setelah agen AI dilatih menggunakan algoritma *Q-Learning* dan *Deep Q-Network (DQN)*, tahap selanjutnya adalah analisis kinerja agen. Pada tahap ini, beberapa langkah akan dilakukan untuk mengevaluasi seberapa baik agen dapat memainkan permainan *Snake*:

- Evaluasi kemampuan agen: Agen AI dilatih melalui berbagai iterasi permainan. Setiap tindakan yang diambil oleh agen menghasilkan *reward* atau penalti, yang digunakan untuk memperbarui strategi agen. Proses ini memungkinkan agen untuk belajar dari pengalaman dan meningkatkan kemampuannya dalam bermain *Snake*.
- Pengumpulan data kinerja: Agen AI dilatih melalui berbagai iterasi permainan. Setiap tindakan yang diambil oleh agen menghasilkan *reward* atau penalti, yang digunakan untuk memperbarui strategi agen. Proses ini memungkinkan agen untuk belajar dari pengalaman dan meningkatkan kemampuannya dalam bermain *Snake*.

- Analisis Hasil: Agen AI dilatih melalui berbagai iterasi permainan. Setiap tindakan yang diambil oleh agen menghasilkan *reward* atau penalti, yang digunakan untuk memperbarui strategi agen. Proses ini memungkinkan agen untuk belajar dari pengalaman dan meningkatkan kemampuannya dalam bermain *Snake*.
- Identifikasi kekuatan dan kelemahan: Agen AI dilatih melalui berbagai iterasi permainan. Setiap tindakan yang diambil oleh agen menghasilkan *reward* atau penalti, yang digunakan untuk memperbarui strategi agen. Proses ini memungkinkan agen untuk belajar dari pengalaman dan meningkatkan kemampuannya dalam bermain *Snake*.

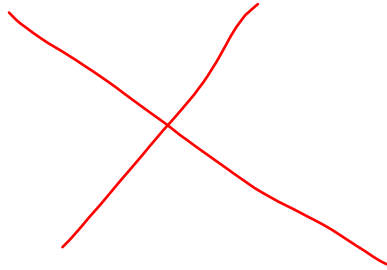
~~Dengan~~ melakukan analisis kinerja yang menyeluruh, diharapkan dapat diperoleh pemahaman yang lebih baik tentang efektivitas metode *Reinforcement Learning* dalam konteks permainan *Snake*, serta potensi pengembangan lebih lanjut untuk aplikasi lain.

3.3.5 Pengujian dan Validasi Hasil

Pengujian dimulai dengan memasukkan agen yang dilatih menggunakan *DQN* ke dalam lingkungan permainan *Snake*. Di sini, agen berinteraksi dengan permainan, mengambil tindakan, dan menerima umpan balik dalam bentuk *reward* berdasarkan seberapa baik agen menghindari dinding, menabrak diri sendiri dan memakan makanan. Proses ini berlangsung dalam beberapa iterasi, dengan tujuan agar agen belajar meningkatkan skornya.

Validasi dilakukan untuk memastikan algoritma *DQN* efektif dan konsisten. Ini dapat dilakukan dengan mengevaluasi performa agen menggunakan metrik seperti total skor atau jumlah makanan yang berhasil dimakan. Selain itu, uji generalisasi dapat dilakukan dengan mengubah beberapa parameter permainan, seperti kecepatan ular atau ukuran papan, untuk memeriksa apakah agen dapat menyesuaikan diri dengan perubahan tersebut.

Setelah pengujian dan validasi, hasilnya dianalisis untuk menilai efektivitas algoritma. Ini mencakup perbandingan kinerja agen *DQN* dengan baseline atau algoritma lain, serta visualisasi proses pembelajaran agen dari waktu ke waktu. Dengan analisis ini, kelemahan dalam algoritma dapat diidentifikasi dan diperbaiki, untuk memastikan apakah agen *DQN* berfungsi optimal dalam permainan *Snake*.



3.4 Jadwal Penelitian

Adapun susunan rencana penelitian yang dilakukan oleh penulis berdasarkan metodologi yang telah dijabarkan sebelumnya. Susunan penelitian dapat dilihat pada Tabel 3.2 sebagai berikut.

Tabel 3.1 Tabel Rencana Penelitian

Tahap	Bulan											
	September				Oktober				November			
	1	2	3	4	1	2	3	4	1	2	3	4
Melakukan studi literasi												
Melakukan simulasi pengembangan lingkungan agen												
Implementasi algoritma												

Header Tabel

Melakukan pelatihan pada agen												
Analisis performa agen												
Melakukan testing dan validasi												
Hasil Analisa pembelajaran agen												
Penulisan laporan												

DAFTAR PUSTAKA

- Alpaydin, E., 2020. *Introduction to Machine Learning*. 4th ed. Cambridge: MIT Press.
- Bostrom, N., 2014. *Superintelligence : paths, dangers, strategies*. 1st ed. Oxford: Oxford : Oxford University Pressv.
- Crespo, J. & Wichert, A., 2020. Reinforcement learning applied to games. *SN Applied Sciences*, Volume 2, p. 824.
- François-Lavet, V. et al., 2018. An Introduction to Deep. *Foundations and Trends in Machine Learning*, 11(3–4), p. 219–354.
- Goodfellow, I., Bengio, Y. & Courville, A., 2016. *Deep Learning*. Massachusetts: MIT Press.
- Harrington, P., 2012. *Machine Learning in Action*v. 1 ed. New York: Manning Publications.
- Kuang, W., 2021. *Fundamentals of Reinforcement Learning*. Edinburg: University of Texas Rio Grande Valley.
- Li, Y., 2018. Deep Reinforcement Learning. *Clinical Orthopaedics and Related Research*, Volume abs/1810.06339.
- Mitchell, T. M., 1997. *Machine Learning*. New York: McGraw-Hill Education.
- Mnih, V. et al., 2015. *Human-level control through deep reinforcement learning*. s.l.:s.n.
- Myerson, R. B., 1991. *Game Theory: Analysis of Conflict*. Cambridge: Harvard University Press.
- Russel, S. & Norvig, P., 2020. *Artificial Intelligence A Modern Approach Fourth Edition*. London: Pearson.

- Silver, D. et al., 2016. Mastering the game of Go with deep neural networks and tree search. *Nature*, Volume 529, p. 484–489.
- Sutton, R. S. & Barto, A. G., 2018. *Reinforcement Learning: An Introduction*. 2nd ed. Cambridge: MIT Press.
- Taqwa, A., Isa, I. G. T. & Ariyanti, I., 2023. Designing A WSNs-based Smart Home Monitoring System through. *Rekayasa Sistem dan Teknologi Informasi*, pp. 1224-1232.
- Tegmark, M., 2017. *Life 3.0 : being human in the age of artificial intelligence*. New York: Alfred A. Knopf.